

图文 13 从磁盘读取数据页到Buffer Pool的时候，free链表有什么用？

手机观看

307 人次阅读 2020-02-07 09:03:31

详情 评论

从磁盘读取数据页到Buffer Pool的时候，free链表有什么用？

如何提问：每篇文章都有评论区，大家可以尽情留言提问，我会逐一答疑

如何加群：购买狸猫技术窝专栏的小伙伴都可以加入狸猫技术交流群，一个非常纯粹的技术交流的地方

具体加群方式，请参见目录菜单下的文档：《MySQL专栏付费用户如何加群》（购买后可见）

1、数据库启动的时候，是如何初始化Buffer Pool的？

现在我们已经搞明白一件事儿了，那就是数据库的Buffer Pool到底长成个什么样，大家想必都是理解了

其实说白了，里面就是会包含很多个缓存页，同时每个缓存页还有一个描述数据，也可以叫做是控制数据，但是我个人是比较倾向于叫做描述数据，或者缓存页的元数据，都是可以的。

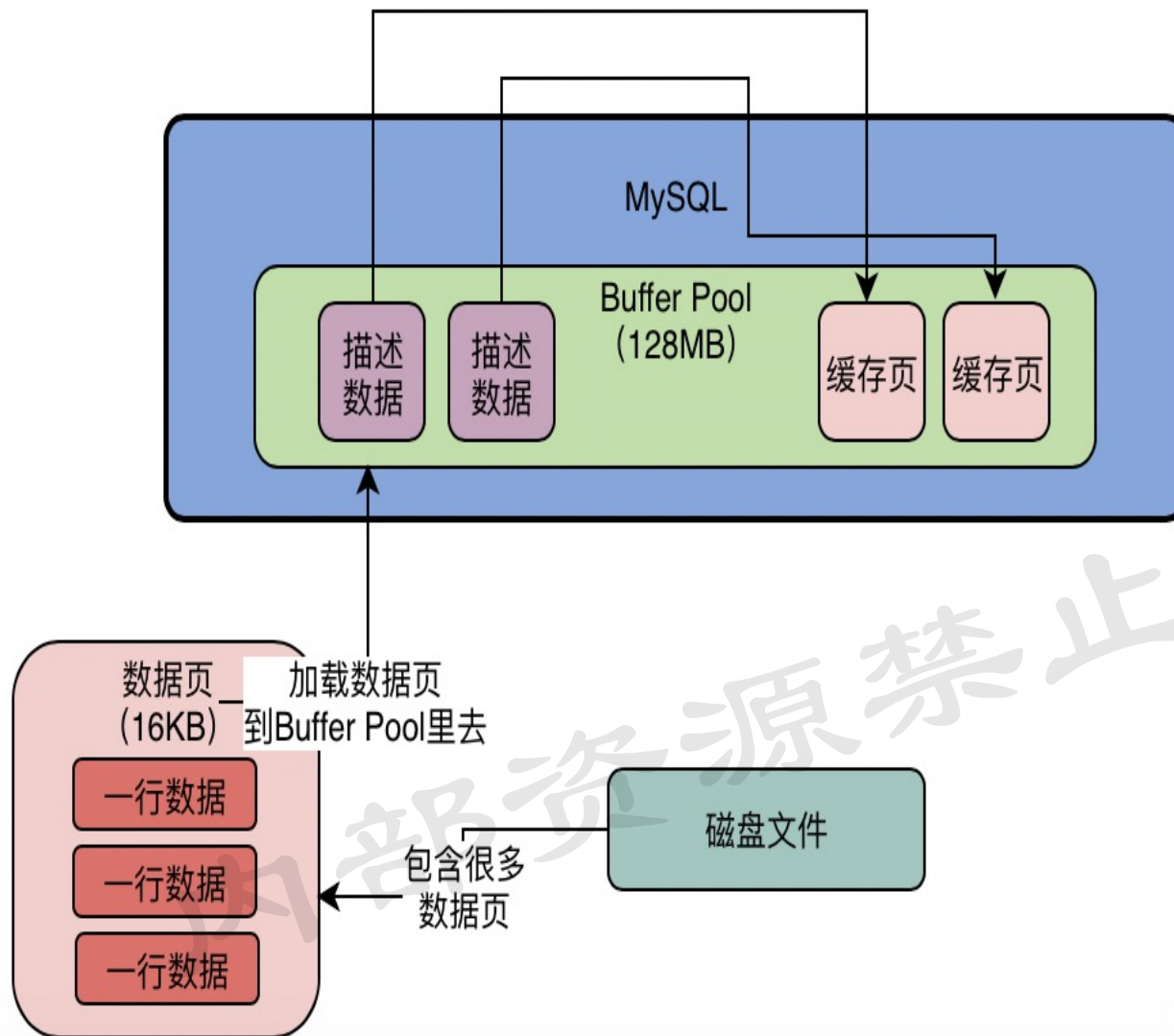
那么在数据库启动的时候，他是如何初始化Buffer Pool的呢？

其实这个也很简单，数据库只要一启动，就会按照你设置的Buffer Pool大小，稍微再加大一点，去找操作系统申请一块内存区域，作为Buffer Pool的内存区域。

然后当内存区域申请完毕之后，数据库就会按照默认的缓存页的16KB的大小以及对应的800个字节左右的描述数据的大小，在Buffer Pool中划分出来一个一个的缓存页和一个一个的他们对应的描述数据。

然后当数据库把Buffer Pool划分完毕之后，看起来就是之前我们看到的那张图了，如下图所示。

内部资源禁止外传



只不过这个时候，Buffer Pool中的一个一个的缓存页都是空的，里面什么都没有，要等数据库运行起来之后，当我们要对数据执行增删改查的操作的时候，才会把数据对应的页从磁盘文件里读取出来，放入Buffer Pool中的缓存页中。

2、我们怎么知道哪些缓存页是空闲的呢？

接着我们来看下一个问题，当你的数据库运行起来之后，你肯定会不停的执行增删改查的操作，此时就需要不停的从磁盘上读取一个一个的数据页放入Buffer Pool中的对应的缓存页里去，把数据缓存起来，那么以后就可以对这个数据在内存里执行增删改查了。

但是此时在从磁盘上读取数据页放入Buffer Pool中的缓存页的时候，必然涉及到一个问题，那就是哪些缓存页是空闲的？

因为默认情况下磁盘上的数据页和缓存页是一一对应起来的，都是16KB，一个数据页对应一个缓存页。

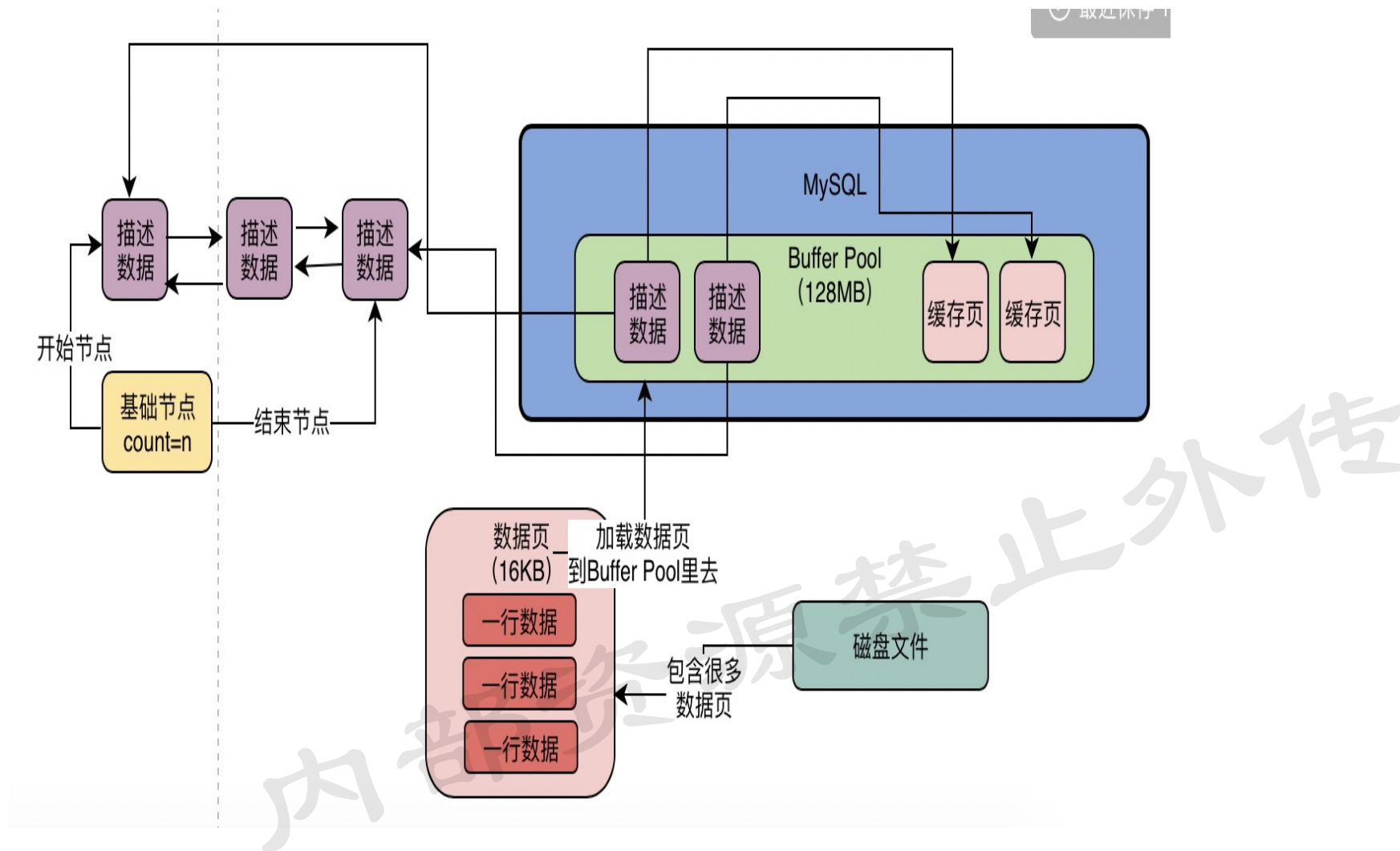
所以我们必须要知道Buffer Pool中哪些缓存页是空闲的状态。

所以数据库会为Buffer Pool设计一个**free链表**，他是一个双向链表数据结构，这个free链表里，每个节点就是一个空闲的缓存页的描述数据块的地址，也就是说，只要你一个缓存页是空闲的，那么他的描述数据块就会被放入这个free链表中。

刚开始数据库启动的时候，可能所有的缓存页都是空闲的，因为此时可能是一个空的数据库，一条数据都没有，所以此时所有缓存页的描述数据块，都会被放入这个free链表中

我们看下图所示

内部资源禁止外传



大家可以看到上面出现了一个free链表，这个free链表里面就是各个缓存页的描述数据块，只要缓存页是空闲的，那么他们对应的描述数据块就会加入到这个free链表中，每个节点都会双向链接自己的前后节点，组成一个双向链表。

除此之外，这个free链表有一个基础节点，他会引用链表的头节点和尾节点，里面还存储了链表中有多少个描述数据块的节点，也就是有多少个空闲的缓存页。

3、free链表占用多少内存空间？

可能有的人会以为这个描述数据块，在Buffer Pool里有一份，在free链表里也有一份，好像在内存里有两个一模一样的描述数据块，是吗？

其实这么想就大错特错了。

这里要给大家讲明白一点，这个free链表，他本身其实就是由Buffer Pool里的描述数据块组成的，你可以认为是每个描述数据块里都有两个指针，一个是free_pre，一个是free_next，分别指向自己的上一个free链表的节点，以及下一个free链表的节点。

通过Buffer Pool中的描述数据块的free_pre和free_next两个指针，就可以把所有的描述数据块串成一个free链表，大家可以自己去思考一下这个问题。上面为了画图需要，所以把描述数据块单独画了一份出来，表示他们之间的指针引用关系。

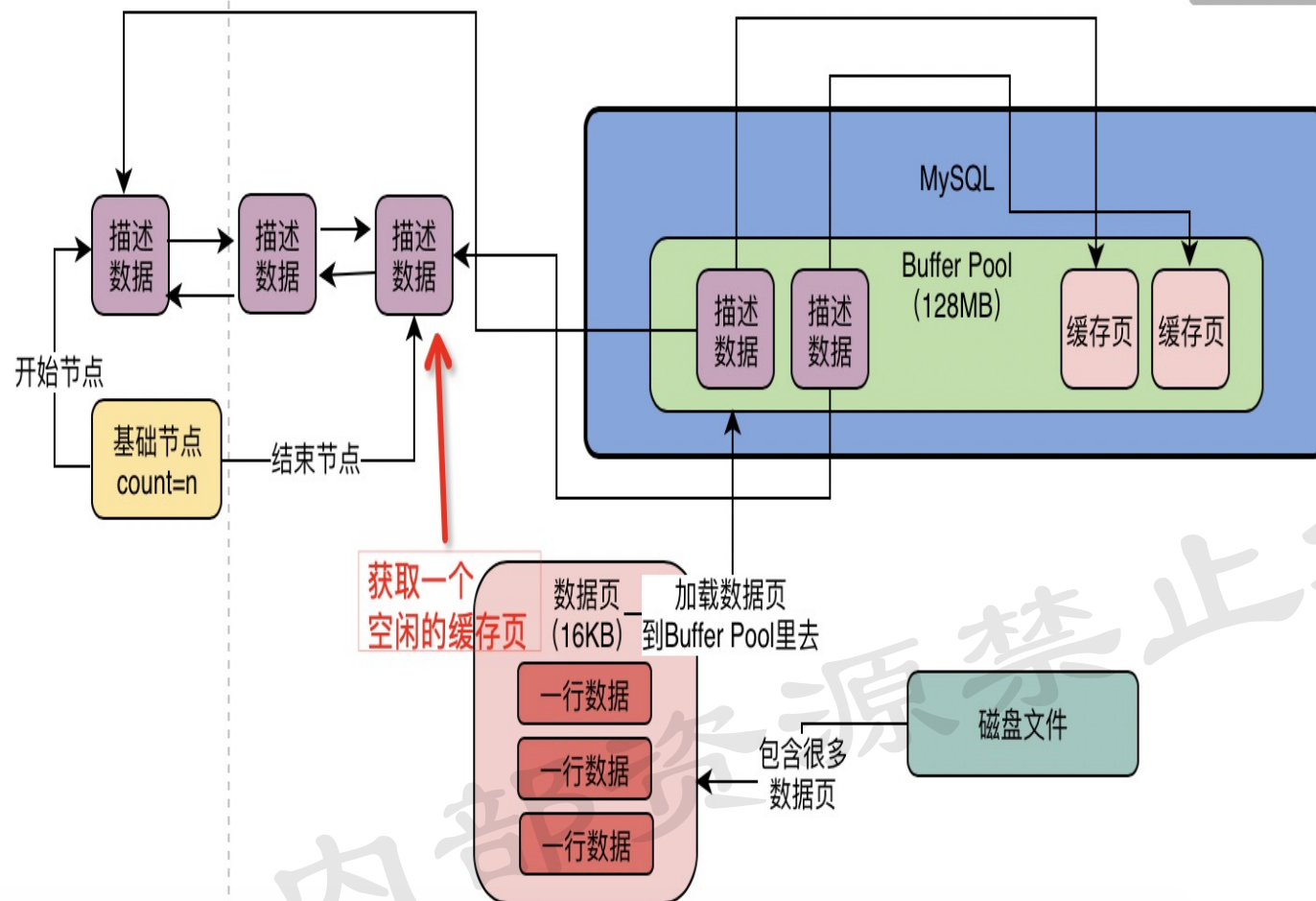
对于free链表而言，只有一个基础节点是不属于Buffer Pool的，他是40字节大小的一个节点，里面就存放了free链表的头节点的地址，尾节点的地址，还有free链表里当前有多少个节点。

4、如何将磁盘上的页读取到Buffer Pool的缓存页中去？

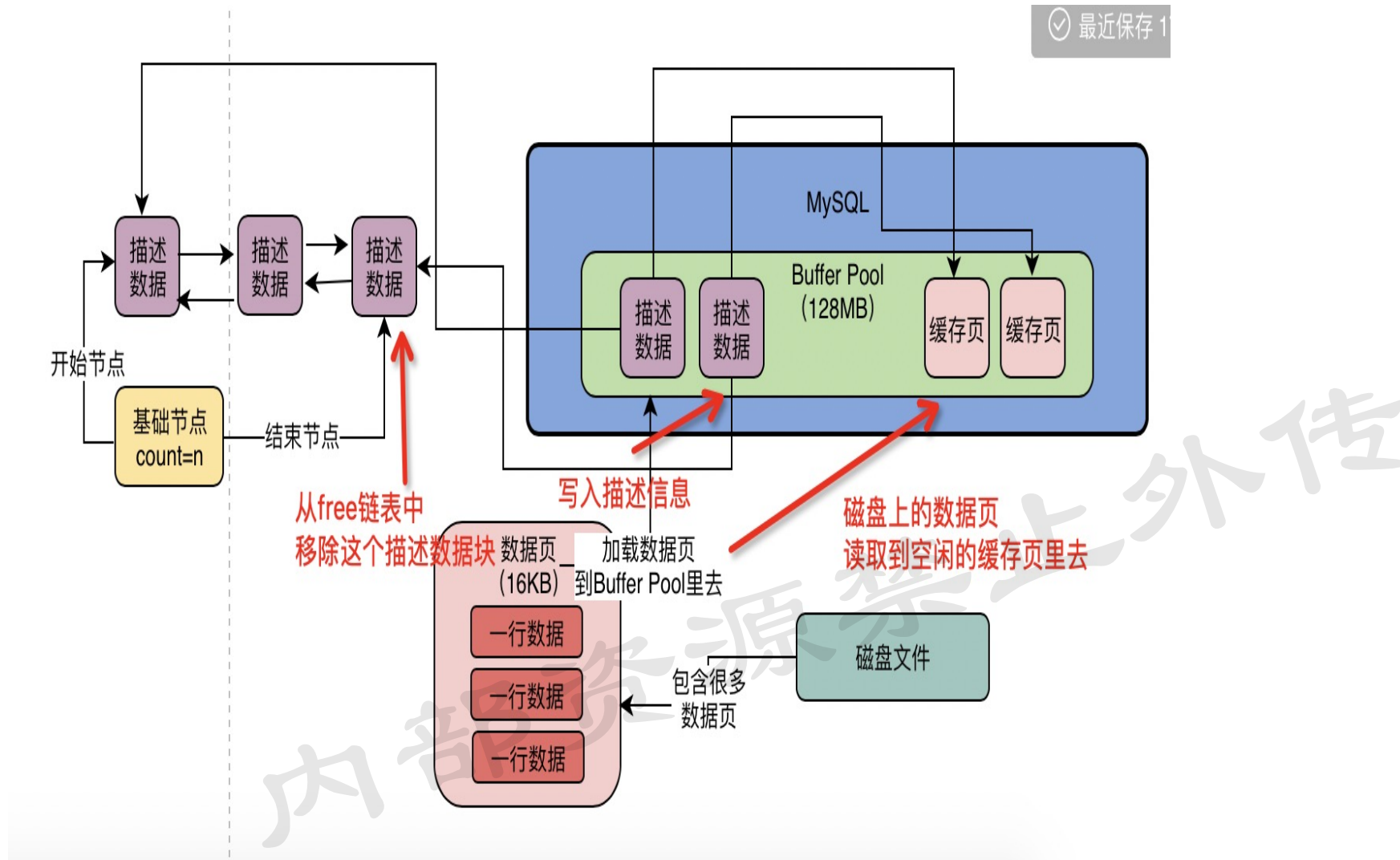
好了，现在我们可以来解答这一篇文章的最后一个问题了，当你需要把磁盘上的数据页读取到Buffer Pool中的缓存页里去的时候，是怎么做到的？

其实有了free链表之后，这个问题就很简单了。

首先，我们需要从free链表里获取一个描述数据块，然后就可以对应的获取到这个描述数据块对应的空闲缓存页，我们看下图所示。



接着我们就可以把磁盘上的数据页读取到对应的缓存页里去，同时把相关的一些描述数据写入缓存页的描述数据块里去，比如这个数据页所属的表空间之类的信息，最后把那个描述数据块从free链表里去除就可以了，如下图所示。



可能有朋友还是疑惑，这个描述数据块是怎么从free链表里移除的呢？

简单，我给你一段伪代码演示一下。

假设有一个描述数据块02，他的上一个节点是描述数据块01，下一个节点是描述数据块03，那么他在内存中的数据结构如下。

```
// 描述数据块
DescriptionDataBlock {
    // 这个块自己就是block02
    block_id = block02
    // 在free链表中的上一个节点是block01
    free_pre = block01;
    // 在free链表中的下一个节点是block03
    free_next = block03;
}
```

现在假设block03被使用了，要从free链表中移除，那么此时直接就可以把block02节点的free_next设置为null就可以了，block03就从free链表里失去引用关系了，如下所示。

```
// 描述数据块
DescriptionDataBlock {
    // 这个块自己就是block02
    block_id = block02
    // 在free链表中的上一个节点是block01
    free_pre = block01;
    // 在free链表中的下一个节点是空的
    free_next = null;
}
```

想必看到这里，大家就完全明白，磁盘中的数据页是如何读取到Buffer Pool中的缓存页里去的了，而且这个过程中free链表是用来干什么的。

5、你怎么知道数据页有没有被缓存？

接着我们来看下一个问题，那你怎么知道一个数据页有没有被缓存呢？

我们在执行增删改查的时候，肯定是先看看这个数据页有没有被缓存，如果没被缓存就走上面的逻辑，从free链表中找到一个空闲的缓存页，从磁盘上读取数据页写入缓存页，写入描述数据，从free链表中移除这个描述数据块。

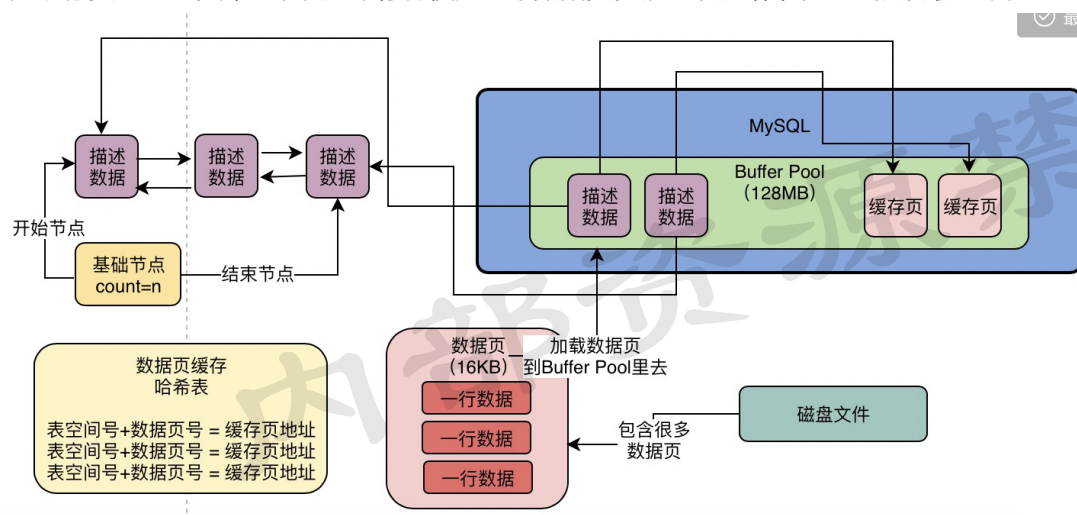
但是如果数据页已经被缓存了，那么就会直接使用了。

所以其实数据库还会有一个哈希表数据结构，他会用表空间号+数据页号，作为一个key，然后缓存页的地址作为value。

当你要使用一个数据页的时候，通过“表空间号+数据页号”作为key去这个哈希表里查一下，如果没有就读取数据页，如果已经有了，就说明数据页已经被缓存了。

我们看下图，又引入了一个数据页缓存哈希表的结构。

也就是说，每次你读取一个数据页到缓存之后，都会在这个哈希表中写入一个key-value对，key就是表空间号+数据页号，value就是缓存页的地址，那么下次如果你再使用这个数据页，就可以从哈希表里直接读取出来他已经被放入一个缓存页了。



6、今日思考题

今天我们给大家留一个思考题，大家去想一个问题，我们要取一个数据的时候，必然会取他所属的一个数据页，而且这个数据必然是属于一个表的，所以我们在上面初步引入了一个表空间的概念

也就是说我们写SQL的时候，只知道表+行的概念，但是在MySQL内部操作的时候，是表空间+数据页的概念。

那么大家觉得这两者之间的区别是什么？他们之间的联系是什么？

请大家积极在评论区写下你的思考，多跟其他人在评论区中交流。

End

专栏版权归公众号**狸猫技术窝**所有

未经许可不得传播，如有侵权将追究法律责任

狸猫技术窝精品专栏及课程推荐：

[《从零开始带你成为消息中间件实战高手》](#)

[《21天互联网Java进阶面试训练营》（分布式篇）](#)

[《互联网Java工程师面试突击》（第1季）](#)

[《互联网Java工程师面试突击》（第3季）](#)

[《从零开始带你成为JVM实战高手》](#)

Copyright © 2015-2020 深圳小鹅网络技术有限公司 All Rights Reserved. 粤ICP备15020529号

 小鹅通提供技术支持